B - Airline Passenger Traffic Prediction

2024-12-16

Step 1: Load and Inspect the Dataset

```
# Load necessary libraries
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.2

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.4.2

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

# Read the data
data <- read.csv("C:\\Users\\USER\\Downloads\\SEASONAL DATA\\international-
airline-passengers.csv")

# Inspect the data
head(data)

##     Month
## 1 1949-01
## 2 1949-02
## 3 1949-03
## 4 1949-04
## 5 1949-05
## 6 1949-06
##
```

```
International.airline.passengers..monthly.totals.in.thousands..Jan.49...Dec.6
0
## 1
112
## 2
118
## 3
132
## 4
129
## 5
121
## 6
135
```

```r
# Check the structure
str(data)
```

```
## 'data.frame':    145 obs. of  2 variables:
##  $ Month
: chr  "1949-01" "1949-02" "1949-03" "1949-04" ...
##  $
International.airline.passengers..monthly.totals.in.thousands..Jan.49...Dec.6
0: int  112 118 132 129 121 135 148 148 136 119 ...
```

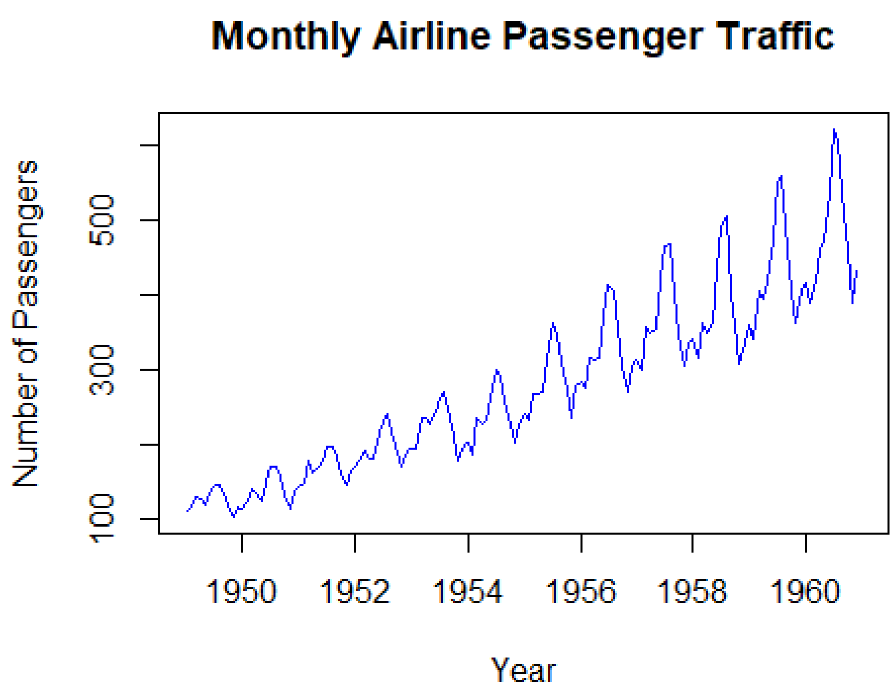
```r
# Summary statistics
summary(data)
```

```
##     Month
##  Length:145
##  Class :character
##  Mode  :character
##
##
##
##
##
International.airline.passengers..monthly.totals.in.thousands..Jan.49...Dec.6
0
##  Min.   :104.0
##  1st Qu.:180.0
##  Median :265.5
##  Mean   :280.3
##  3rd Qu.:360.5
##  Max.   :622.0
##  NA's   :1
```

Step 2: Convert to Time Series Object

```r
# Rename the column for clarity
colnames(data) <- c("Month", "Passengers")
```

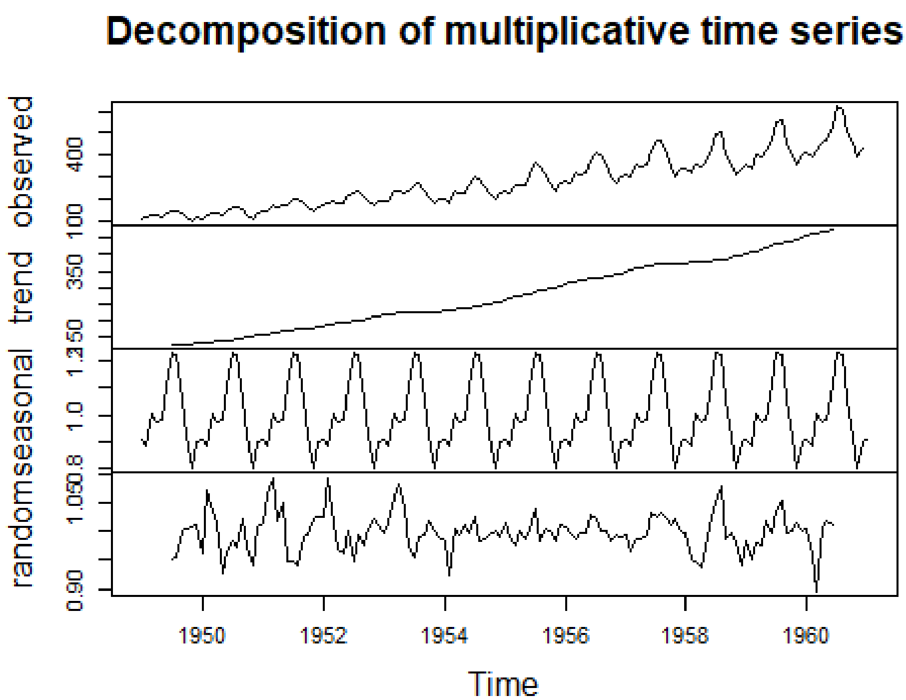
```r
# Convert 'Month' to Date format
data$Month <- as.Date(data$Month, format = "%Y-%m")

# Create the time series object
passenger_ts <- ts(data$Passengers, start = c(1949, 1), frequency = 12)

# Plot the time series
plot(passenger_ts, main = "Monthly Airline Passenger Traffic",
     xlab = "Year", ylab = "Number of Passengers", col = "blue")
```




```r
# Check for missing values
sum(is.na(data$Passengers))

## [1] 1

# Ensure the Passengers column is numeric
data$Passengers <- as.numeric(data$Passengers)

# Verify the structure
str(data)

## 'data.frame':    145 obs. of  2 variables:
##  $ Month     : Date, format: NA NA ...
##  $ Passengers: num  112 118 132 129 121 135 148 148 136 119 ...
```

Step 3: Decompose the Time Series

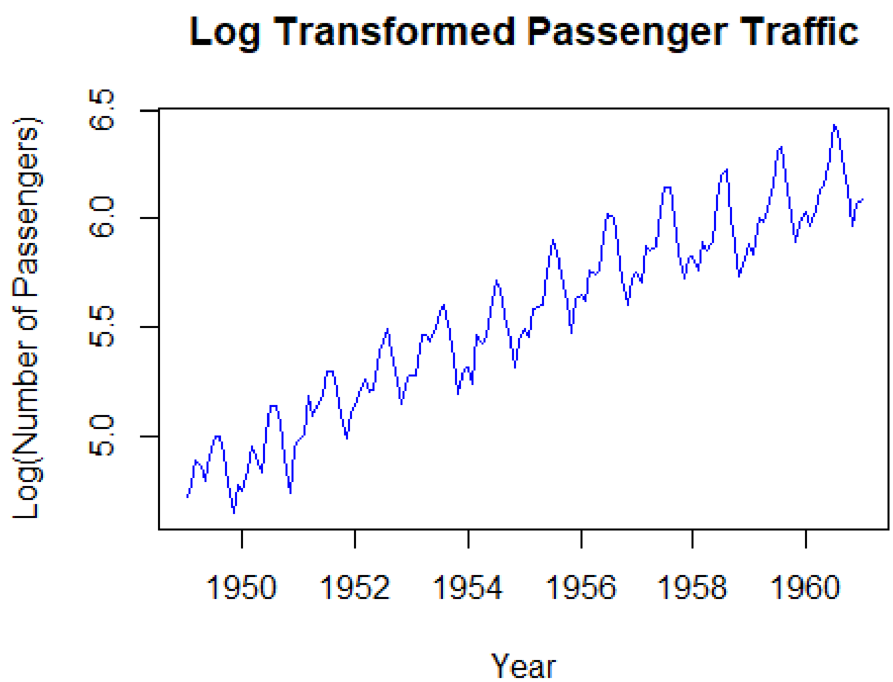
```r
# Decompose the time series
decomposed <- decompose(passenger_ts, type = "multiplicative")

# Plot the decomposition
plot(decomposed)
```



Step 4: Check for Stationarity

```r
# Check for zeros or NAs in the time series
sum(is.na(passenger_ts))   # Count NA values

## [1] 1

sum(passenger_ts == 0)      # Count zero values

## [1] NA

# Interpolate missing values
library(forecast)

## Warning: package 'forecast' was built under R version 4.4.2

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

passenger_ts <- na.interp(passenger_ts)
```

```r
# Apply log transformation
log_passenger_ts <- log(passenger_ts)

# Verify the log-transformed series
plot(log_passenger_ts, main = "Log Transformed Passenger Traffic",
     xlab = "Year", ylab = "Log(Number of Passengers)", col = "blue")
```




```r
# Perform ADF test
library(tseries)

## Warning: package 'tseries' was built under R version 4.4.2

adf_test <- adf.test(log_passenger_ts, alternative = "stationary")

## Warning in adf.test(log_passenger_ts, alternative = "stationary"): p-value
## smaller than printed p-value

print(adf_test)

##
##  Augmented Dickey-Fuller Test
##
## data:  log_passenger_ts
## Dickey-Fuller = -6.3982, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

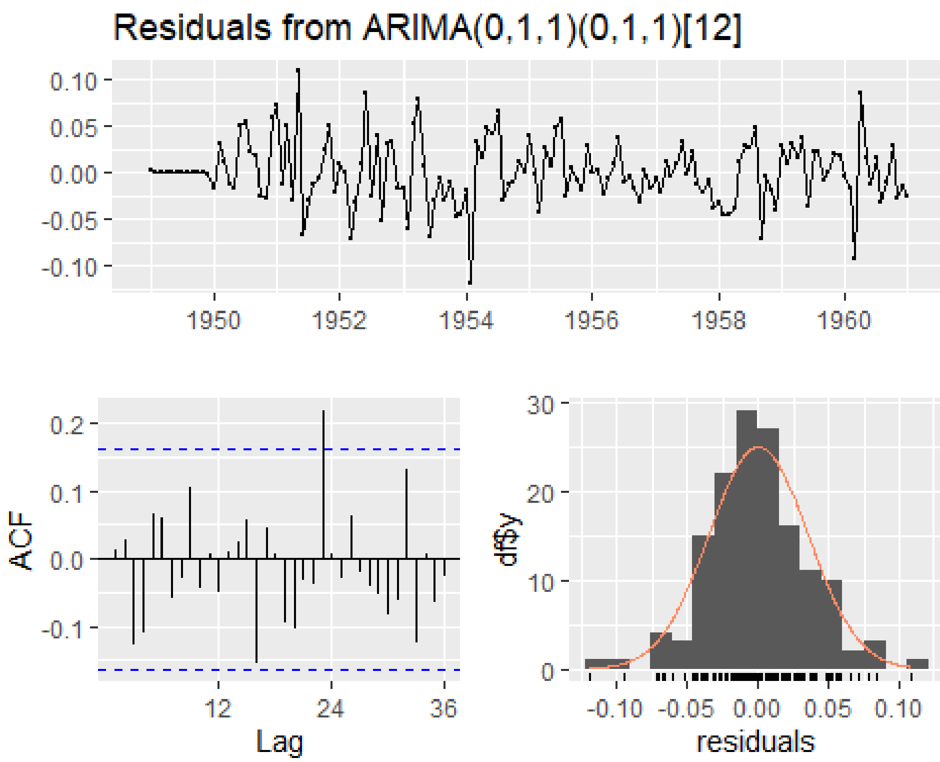
Step 5: Fit a SARIMA Model

```r
# Load the forecast library
library(forecast)

# Fit the SARIMA model
sarima_model <- auto.arima(log_passenger_ts, seasonal = TRUE)

# Print the summary of the fitted model
summary(sarima_model)
```

```
## Series: log_passenger_ts
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##           ma1     sma1
##       -0.3983  -0.5577
## s.e.   0.0895   0.0730
##
## sigma^2 = 0.001366:  log likelihood = 246.84
## AIC=-487.68   AICc=-487.49   BIC=-479.03
##
## Training set error measures:
##                        ME       RMSE        MAE         MPE      MAPE
MASE
## Training set 0.0003871913 0.03499213 0.02626068 0.007912188 0.4749857
0.2179003
##                    ACF1
## Training set 0.01347755
```

Step 6: Check Model Diagnostics

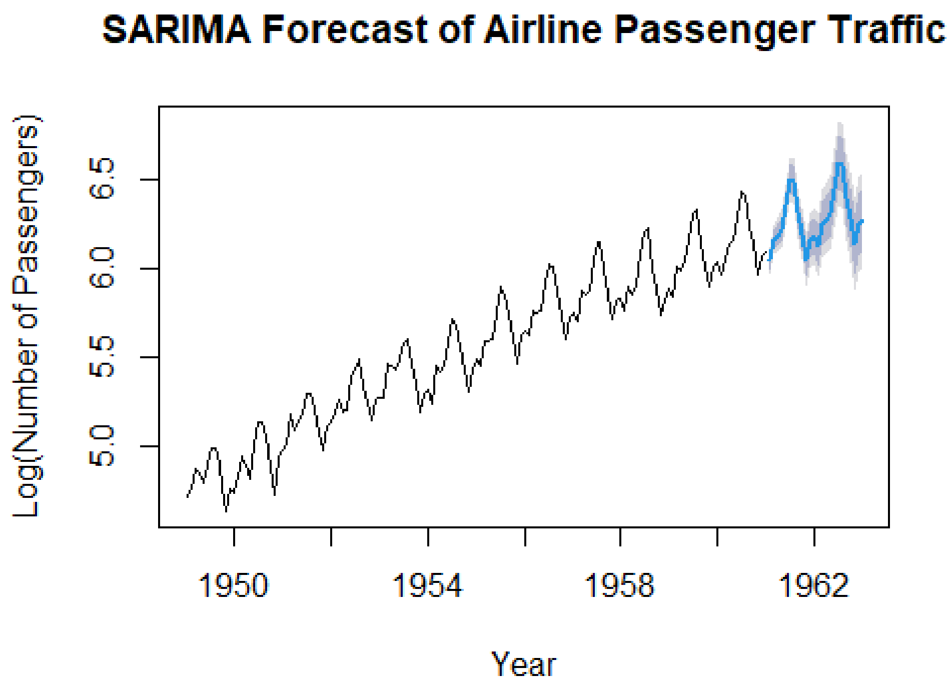
```r
# Plot residual diagnostics
checkresiduals(sarima_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 25.798, df = 22, p-value = 0.2605
##
## Model df: 2.   Total lags used: 24
```

Step 7: Forecast Future Values

```
# Forecast the next 24 months
forecast_values <- forecast(sarima_model, h = 24)

# Plot the forecast
plot(forecast_values, main = "SARIMA Forecast of Airline Passenger Traffic",
     xlab = "Year", ylab = "Log(Number of Passengers)")
```




```r
# Convert forecast back to the original scale
forecast_values$mean <- exp(forecast_values$mean)

# Print forecasted values
print(forecast_values)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 1961       419.1327 5.990827 6.085548 5.965755 6.110620
## Mar 1961       471.6275 6.100917 6.211462 6.071657 6.240721
## Apr 1961       484.7735 6.121496 6.245867 6.088577 6.278787
## May 1961       501.1607 6.148523 6.285331 6.112312 6.321541
## Jun 1961       574.3120 6.279071 6.427275 6.239844 6.466502
## Jul 1961       659.6126 6.412261 6.571045 6.370233 6.613072
## Aug 1961       656.7351 6.402930 6.571632 6.358277 6.616284
## Sep 1961       549.5520 6.220069 6.398138 6.172937 6.445270
## Oct 1961       489.4998 6.099901 6.286867 6.050414 6.336354
## Nov 1961       423.2187 5.950159 6.145619 5.898424 6.197354
## Dec 1961       469.8616 6.050639 6.254238 5.996749 6.308127
## Jan 1962       482.6952 6.073673 6.285098 6.017712 6.341059
## Feb 1962       458.3117 6.010846 6.244253 5.949067 6.306032
## Mar 1962       515.7136 6.121823 6.369280 6.056325 6.434778
## Apr 1962       530.0884 6.142668 6.403420 6.073651 6.472436
## May 1962       548.0075 6.169589 6.442989 6.097224 6.515354
## Jun 1962       627.9967 6.299790 6.585280 6.224226 6.660844
## Jul 1962       721.2709 6.432471 6.729558 6.353837 6.808192
## Aug 1962       718.1244 6.422519 6.730767 6.340930 6.812355
## Sep 1962       600.9222 6.238956 6.557975 6.154517 6.642414
```

```
## Oct 1962       535.2565 6.118027 6.447465 6.030830 6.534662
## Nov 1962       462.7797 5.967482 6.307020 5.877612 6.396890
## Dec 1962       513.7826 6.067127 6.416473 5.974661 6.508939
## Jan 1963       527.8159 6.089305 6.448190 5.994313 6.543181
```